

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

U.S. PATENT APPLICATION

FOR:

MOBILE WEB SERVICES

INVENTOR:

PETRI NYKÄNEN

Morgan & Finnegan L.L.P.

345 Park Avenue, 22nd Floor

New York, NY 10154-0053

(212) 758-4800 (Telephone)

(212) 751-6849 (Facsimile)

1775 Eye Street, N.W., Suite 400

Washington D.C., 20006

(202) 857-7887 (Telephone)

(202) 857-7929 (Facsimile)

Attorneys For Applicant

MOBILE WEB SERVICES

5 BACKGROUND OF THE INVENTION

Field of the Invention:

The invention disclosed broadly relates to methods for providing Internet services and more particularly relates to improvements in mobile device accessing of Internet services.

Background Art:

Universal Description, Discovery and Integration (UDDI) is a defacto standard for an Internet-based registry. The UDDI registry enables users to discover services and businesses on the Internet. The UDDI standard takes advantage of WorldWide Web Consortium (W3C) standards and Internet Engineering Task Force (IETF) standards such as Extensible Markup Language (XML), Hypertext Transfer Protocol (HTTP), Domain Name System (DNS) protocol, and Simple Object Access Protocol (SOAP) messaging protocol. The UDDI registry enables users to quickly, easily and dynamically find businesses and services on the Internet. The UDDI registry enables businesses to reach their customers and partners with information about their products and Web services. The UDDI registry also enables businesses to integrate into each other's systems and processes. Registering enables a business to publicly list basic information about its company and offerings. There is also the option to list a catalog of products, services and guidelines for engagement. Registered businesses and their catalogs of services and products

are then accessible in searches by potential buyers. Details of the UDDI registry and its searching protocol can be found in the following online papers:

UDDI Executive White Paper, September 6, 2000,

http://www.uddi.org/pubs/UDDI_Executive_White_Paper.pdf

UDDI Technical White Paper, September 6, 2000,

http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf

UDDI Programmer's API 1.0, UDDI Open Draft Specification 30 September 2000, by

Toufic Boubez, et al., <http://www.uddi.org/pubs/ProgrammersAPI-V1-1.pdf>

UDDI Data Structure Reference, UDDI Open Draft Specification 30 September 2000, by

Toufic Boubez, et al., <http://www.uddi.org/pubs/DataStructure-V1.pdf>

Mobile phones and wireless personal digital assistants (PDAs) are able to access the Internet using the Wireless Application Protocol (WAP). The Nokia WAP Client Version 2.0 is a software product containing the components necessary to implement a WAP client on a wireless device. These components include a Wireless Markup Language (WML) Browser, WMLScript engine, Push Subsystem, and Wireless Protocol Stack. The Nokia WAP Client is a source-code product that can port and integrate into wireless devices such as mobile phones and wireless PDAs. Application programs stored in the wireless device interact with the WAP Client to implement a variety of communications applications. Details of the Nokia WAP Client

Version 2.0 can be found in the online paper: Nokia WAP Client Version 2.0, Product Overview,
Nokia Internet Communications, 2000, www.nokia.com/corporate/wap.

What is needed is the ability of a mobile phone or wireless PDA to discover Internet
businesses and services by accessing the UDDI registry. It would be even more useful to
5 facilitate the formation of a query to the UDDI registry for the wireless device user. It would be
beneficial to maintain a personal profile of the user's Internet accessing preferences and to use
that profile in formulating a detailed UDDI query of the UDDI registry from the user's
abbreviated inputs to the wireless device.

SUMMARY OF THE INVENTION:

In accordance with the invention, a method is disclosed to enable a mobile phone or
wireless PDA to discover Internet businesses and services by accessing the Universal
Description, Discovery and Integration (UDDI) registry. The method facilitates the formation of
15 a query to the UDDI registry for the wireless device user. The method constructs a personal user
profile of the user's UDDI searching strategies and Internet accessing preferences. The user
profile can be used as a shortcut for online or offline queries to the UDDI registry or for
accessing pages from web sites, in response to the user's entry of abbreviated inputs to the
wireless device. The method is embodied as programmed instructions which may be executed
20 within the user's wireless device to query the UDDI registry. Alternately, method is embodied as
programmed instructions which may be executed within a separate knowledge engine server to
query the UDDI registry in response to commands from the user's wireless device. The server

can be used to cache files accessed from web sites, for selective forwarding to the user's wireless device.

In one aspect of the invention, the sequence of operational steps for the wireless device's UDDI registry browsing method begins by entering a search handle that will be associated with the user's search strategy. Then the query terms are entered by the user, which may be a business name, part of a business name, a service description, or other characterization. If the characterization is part of a business name, the wireless device then sends a *find_business* XML inquiry using Simple Object Access Protocol (SOAP) to the UDDI registry. The wireless device then receives back from the UDDI registry, a *businessList* message that contains a list of business names satisfying the *find_business* query. The user can then select an item from the returned *businessList* message and drill-down in the selected business' entity data.

The wireless device then sends a *find_service* XML inquiry using SOAP to the UDDI registry. The wireless device then receives back from the UDDI registry, a *serviceList* message that contains a list of names of services offered by the selected business. The user can then select an item from the returned *serviceList* message and drill-down in the selected service data.

The wireless device then sends a *_get_serviceDetail_* XML inquiry using SOAP to the UDDI registry. The wireless device then receives back from the UDDI registry, a *serviceDetail* message that includes *bindingTemplate* data that contains the details of the selected service. Included in the *bindingTemplate* data is the *accessPoint URL*, which is the URL of the selected service on the web site of the selected business. The service details may be comprehensive or abbreviated, and may also be prioritized according to the needs of the user (or not prioritized at all).

The wireless device then displays the accessPoint URL to the user. The wireless device also stores the search handle in user profile with the selected accessPoint URL, to enable the user to access web pages from the web site of the selected business. This provides the user with a shortcut for accessing pages from web sites, in response to the user's entry of abbreviated search handle to the wireless device.

The wireless device also stores the search handle in user profile with the UDDI registry search strategy. The stored search strategy includes the business name query , the selected *businessEntity* data, the selected *businessService* data, the selected *bindingTemplate* data, and the selected *accessPoint* URL. This provides the user with a shortcut for online or offline queries to the UDDI registry, in response to the user's entry of abbreviated search handle to the wireless device. The search handle may be associated with an icon on the user's screen. Thus, activation of the icon by the user would initiate the shortcut.

To replay a UDDI registry search strategy, the user merely enters a search handle into the wireless device and selects the replay option. The wireless device then accesses the UDDI registry search strategy from user profile corresponding to the search handle and loads the business name query , the selected *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data as each respective operand that would have been otherwise entered by the user. If the data in the UDDI registry has been updated since the user's last query, the *bindingTemplate* data may contain additional or modified *accessPoint URLs*, of the selected service on the web site of the selected business.

In another aspect of the invention, the sequence of operational steps described above for the wireless device's UDDI registry browsing method can also be carried out in a separate knowledge engine server. The server performs the above described method to query the UDDI

registry in response to commands from the user's wireless device. The knowledge engine in the server begins by receiving user's query or search handle. If a search handle has been received, then the server replays a corresponding search strategy for the UDDI registry accessed from the user's profile and uses the query steps specified in the strategy instead of requesting further inputs from the user.

If, however, the knowledge engine server has received a new user query, the server then accesses the UDDI registry using the method described above to identify web sites. The server returns a list of web sites to the user and stores binding templates in the user's profile. The server then receives the user's selection of web sites to search and the server updates user profile with these preferences.

Whether the server begins by receiving the user's new query or the user's search handle, the server proceeds to search the identified web sites using the URLs contained in the stored binding templates. The server retrieves any documents resulting from the search of the specified web sites. The server applies a filter that is prescribed by the user and stored in the user's profile, to limit the returned documents to only those of particular interest to the user. The server sorts the documents in a list having an order established in accordance with user's profile. The server further stores the filtered documents and the sorted list in a cache for later use. The documents may subsequently be accessed selectively by the user. The server also returns the list of documents to user, and if the user is not logged on, the list will be returned to the user when he/she next logs on. It should be noted that the filter may be automatically created, based on the profile information or history information available on the interests of the user. Thus it would be possible, for example, for a server to determine interests of a user through earlier selections that were made.

The server receives the user's selections from the list and it updates the user's profile with the user's preferences. The server then returns the selected documents to user. As was described above, the knowledge engine server associates the search handle with user's selections and with the user's search strategy, storing that association in user's profile. It is understood that the term "document" under the present invention relates not just to web page documents, but also to services such as streaming video, audio or other application-specific communication. Thus, the present disclosure is not limited to just browsing after user discovery has taken place.

DESCRIPTION OF THE FIGURES:

Figure 1 is a network diagram of the invention, showing an exemplary relationship between the user's Wireless Application Protocol (WAP)-enabled portable wireless device, the WAP protocol gateway to the Internet, the knowledge engine server, the Universal Description, Discovery and Integration (UDDI) registry, and a plurality of web sites;

Figure 1A through 1H show the user's wireless device in a progression of stages as it carries out the UDDI registry browsing method;

Figure 1I and 1J show the user's wireless device in a progression of stages as it carries out the method of replaying a UDDI registry search strategy as a shortcut for online or offline queries to the UDDI registry;

Figure 2 is a functional block diagram of the wireless device 100, showing its various components and programs;

Figure 2A is a flow diagram of the sequence of operational steps for the wireless device's UDDI registry browsing program;

Figure 2B is a flow diagram of the sequence of operational steps for the wireless device's program to replay the UDDI registry search strategy;

Figure 3A is a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out the UDDI registry browsing program of **Figure 2A**;

Figure 3B is a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out the program to replay the UDDI registry search strategy;

Figure 4 is a functional block diagram of the knowledge engine server, showing the memory storing the application services software programs needed to perform the operations of the invention;

Figure 4A is a more detailed functional block diagram of the server, showing the knowledge engine;

Figure 4B is a network process flow diagram of the interaction of the wireless device, the knowledge engine server, and the UDDI registry when carrying out the UDDI registry browsing program in the server.

DISCUSSION OF THE PREFERRED EMBODIMENT:

The invention is applied to wireless telephones and wireless personal digital assistants (PDAs) implementing the Wireless Application Protocol (WAP) standard. **Figure 1** is a network diagram of an embodiment of the invention, illustrating a relationship between the user's Wireless Application Protocol (WAP)-enabled portable wireless device **100**, a WAP protocol gateway **120**, and the server **140**. The user's WAP-enabled portable wireless device **100** can be a wireless mobile phone, pager, two-way radio, smartphone, personal communicator, or the like. The user's WAP-enabled portable wireless device **100** accesses a small file called a deck which is composed of several smaller pages called cards which are small enough to fit into the display area of the device's microbrowser **102**. The small size of the microbrowser **102** and the small file sizes accommodate the low memory constraints of the portable wireless device **100** and the low-bandwidth constraints of a wireless network **116**. The cards are written in the Wireless Markup Language (WML) which is specifically devised for small screens and one-hand navigation without a keyboard. The WML language is scaleable from two-line text displays on the microbrowser **102** of a cellular telephone, up through large LCD screens found on smart phones and personal communicators. The cards written in the WML language can include programs written in WMLScript, which is similar to JavaScript, but makes minimal demands on memory

and CPU power of the device 100 because it does not contain many of the unnecessary functions found in other scripting languages.

The Nokia WAP Client Version 2.0 is a software product containing the components necessary to implement a WAP client on a wireless device. These components include a Wireless Markup Language (WML) Browser, WMLScript engine, Push Subsystem, and Wireless Protocol Stack. The Nokia WAP Client is a source-code product that can port and integrate into wireless devices such as mobile phones and wireless PDAs. Application programs stored in the wireless device interact with the WAP Client to implement a variety of communications applications. Details of the Nokia WAP Client Version 2.0 can be found in the online paper: Nokia WAP Client Version 2.0, Product Overview, Nokia Internet Communications, 2000, www.nokia.com/corporate/wap. It is also understood that the system and methods disclosed herein are applicable to other platforms as well, such as XHTML, and is not limited strictly to the WAP protocol.

The user's WAP-enabled portable wireless device 100 communicates with the radio tower 114 and can exchange messages for distances up to several kilometers. The types of wireless networks 116 supported by the WAP standard include Cellular Digital Packet Data (CDPD), Code-Division Multiple Access (CDMA), Global System for Mobile Communications (GSM), Time Division Multiple Access (TDMA), GPRS, 3G-Broadband, and the like.

The overall process of communication between the user's WAP-enabled wireless device (the client) 100, through the WAP protocol gateway 120, to the server 140 resembles the way Web pages are served on the Internet using the HyperText Transfer Protocol (HTTP) or World Wide Web protocol:

[1] The user presses a key on the user's device **100** related to the Uniform Resource Locator (URL) of the server **140**.

[2] The user's device **100** sends the URL, via the radio tower **114** and the wireless network **116**, to the gateway **120** using WAP protocols.

[3] The gateway **120** translates the WAP request into an HTTP request and sends it over the Internet **130** to the server **140**, via Transmission Control Protocol/ Internet Protocol (TCP/IP) interfaces.

[4] The server **140** handles the request just like any other HTTP request received over the Internet. The server **140** either returns a WML deck or a HyperText Markup Language (HTML) page back to the gateway **120** using standard server programs written, for example in Common Gateway Interface (CGI) programs, Java servlets, or the like.

[5] The gateway **120** receives the response from the server **140** on behalf of the user's device **100**. If the response is an HTML page, it gets transcoded into WML if necessary. Then the WML and WMLScript coding is encoded into a byte code that is then sent to the user's device **100**.

[6] The user's device **100** receives the response in the WML byte code and displays the first card in the deck on the microbrowser **102** to the user.

In **Figure 1**, the protocol gateway **120** includes a WAP protocol stack organized into five different layers (not shown). An application layer is the wireless application environment, which executes portable applications and services. A session layer is the wireless session protocol, which supplies methods for the organized exchange of content between client/server applications. A transaction layer is the wireless transaction protocol, which provides methods for

performing reliable transactions. A security layer is the wireless transport layer security, which provides authentication, privacy, and secure connections between applications. The transport layer is the wireless datagram protocol, which shelters the upper layers from the unique requirements of the diverse wireless network protocols, such as CDPD, CDMA, GSM, etc.

5 Additional information about the WAP standard and the WAP protocol stack can be found in the book by Charles Arehart, et al. entitled, "Professional WAP", published by Wrox Press Ltd., 2000 (ISBN 1-861004-04-1).

In **Figure 1**, the user's portable wireless device **100** includes the microbrowser **102** displaying the Mobile Web Services menu, that enables the user to navigate through the cards being displayed and to select options that are programmed by the application programs **106**. The user's device **100** also includes the WAP client program **108** which has been previously discussed.

The Mobile Web Services menu displayed by the microbrowser **102** in **Figure 1** is rendered by the WAP client program **108** under the control of the application programs **106**, which are further illustrated in **Figures 2, 2A, and 2B**. The user can select the session type utilizing the Mobile Web Services menu, by either [A] a direct session with the UDDI registry or [B] an indirect session with the UDDI registry through a knowledge server **140**. The direct session with the UDDI registry is further illustrated in the network process flow diagrams of **Figures 3A and 3B**. The knowledge service adds value to UDDI searches with pre and post-processing, regardless of the device being used to accesss UDDI. The indirect session with the UDDI registry through the knowledge server **140** is further illustrated in the network process flow diagram of **Figure 4B**. Whichever session type is selected by the user, the Mobile Web

Services menu of **Figure 1** presents to the user the UDDI Registry Search Menu from which the user can select the following options:

[1] BROWSE UDDI REGISTRY FOR WEB SITE URLS

[2] DRILL-DOWN UDDI DATA FOR DETAILS

[2A] BUSINESS ENTITY DATA

[2B] BUSINESS SERVICE DATA

[2C] BINDING TEMPLATE DATA

[2D] T_MODEL DATA

[3] INVOKE WEB SITE W/ CACHED UDDI DATA

[4] ENTER SEARCH HANDLE TO USE
PROFILE FROM A PRIOR SEARCH

Option [1] of browsing the UDDI registry for web site URLs allows the user to explore and examine data organized by the UDDI registry in a hierarchy. The core information model used by the UDDI registries is defined in an XML schema. XML allows hierarchical relationships to be described for four types data: business information; service information, binding information; and information about specifications for services.

A first type of data is Business information, which is specified in the UDDI registry with the *businessEntity* XML element. The *businessEntity* XML element typically includes the name, general description, contacts, and categories of the business whose web site is on the Web. The *businessEntity* XML element serves as the top of the information hierarchy for all of the information about a business under the present embodiment.

A second type of data is Service information, which is specified in the UDDI registry with the *businessService* XML element. One or more *businessService* XML elements is contained in each *businessEntity* XML element. The *businessService* XML element includes one

or more *bindingTemplate* XML elements, which are a third type of data. The *businessService* XML element is a descriptive container that is used to group a series of related Web services related to either a business process or category of services, such as purchasing services, shipping services, news services, and other high-level business processes. The *businessService* XML element information sets can each be further categorized in combinations of industry, product and service or geographic subjects.

The *bindingTemplate* XML element contains the detailed technical descriptions of Web services. As such, these structures provide the technical entry point URL for specific services and products offered by a business. Each *bindingTemplate* XML element structure has a single logical *businessService* XML element parent, which in turn has a single logical *businessEntity* XML element parent. An important piece of information in the *bindingTemplate* XML element is the *accessPoint* element. The *accessPoint* element is the URL of a specific service provided by the business. A single attribute is typically provided, and is identified in the present embodiment as *URLType*. The purpose of the *URLType* attribute is to facilitate searching for entry points associated with a particular type of service. An example might be a purchase order service that provides three entry points, one for HTTP, one for SMTP, and one for FAX ordering. In this example, a *businessService* XML element contains three *bindingTemplate* XML element entries, each with identical data with the exception of the *accessPoint* value and *URLType* value.

A fourth type of data in the UDDI registry is the *tModel* XML element, which is pointed to by a pointer in the *bindingTemplate* XML element. The *tModel* XML element specifies the protocols, interchange formats and interchange sequencing rules for accessing web pages from the business' server having the service information specified in the *businessService* XML element.

Option [1] of the Mobile Web Services menu of **Figure 1**, is:

[1] BROWSE UDDI REGISTRY FOR WEB SITE URLS

Option [1] of browsing the UDDI registry for web site URLs involves starting with some broad information, performing a search, finding general result sets and then selecting more specific information for drill-down. The UDDI registry accommodates the browse pattern with the *find_xx* API call. These calls form the search capabilities provided by the UDDI registry and are matched with return messages that return overview information to match the supplied search criteria. A typical browse sequence may involve finding whether a particular business has any information registered in the UDDI registry. This sequence would start with a call *find_business*, and may subsequently pass the first few characters of the businesses name. The UDDI registry would then return a *businessList* result. The *businessList* result is a list of overview information (keys, names and description) of the *businessEntity* information that matched the search results returned by the *find_business* query. **Figure 1A** through **1H** illustrate the user's wireless device in a progression of stages as it carries out the UDDI registry browsing method. **Figure 2A** illustrates a flow diagram of the sequence of operational steps for the wireless device's UDDI registry browsing program. **Figure 3A** illustrates a network process flow diagram, showing the interaction of the wireless device and the UDDI registry when carrying out the UDDI registry browsing program of **Figure 2A**.

Option [2] of the Mobile Web Services menu of **Figure 1**, is:

[2] DRILL-DOWN UDDI DATA FOR DETAILS

[2A] BUSINESS ENTITY DATA
[2B] BUSINESS SERVICE DATA
[2C] BINDING TEMPLATE DATA
[2D] T_MODEL DATA

5

After the user has identified a business he/she has been browsing for in Option [1], the user can drill-down into their *businessService* information. Here, the user may search for particular service types (e.g. purchasing, shipping, news and the like) using the *find_service* API call. Similarly, if the user knows the technical fingerprint (*tModel* signature) of a particular product and wants to see if the business he/she has chosen supports a compatible service interface, the user can use the *find-binding* API call.

Once the user has a key for one of the four main data types managed by the UDDI registry, he/she can use that key to access the full registered details for a specific data type (*businessEntity*, *businessService*, *bindingTemplate*, or *tModel*). The full registered information for any of these structures can be accessed by passing a relevant key type to one of the *get_xx* API calls to the UDDI registry.

Continuing with the example on browsing, one of the data items returned by all of the *findL-xx* return sets is key information. In the case of a business that the user is interested in, the *businessKey* value returned within the contents of a *businessList* structure can be passed as an argument to the UDDI registry to *get_businessDetail*. The successful return of this message is a *businessDetail* message containing the full registered information for the entity whose key value was passed. This typically will be a full *businessEntity* structure. Since full *businessEntity* structures can contain a large quantity of information, it can be optionally cached in the cache of the knowledge engine server 140 of Figure 1.

25

Option [3] of the Mobile Web Services menu of **Figure 1**, is:

[3] INVOKE WEB SITE W/ CACHED UDDI DATA

After the user has retrieved the *businessEntity* structure from the UDDI registry **170** of **Figure 1**, the user can access the *accessPoint* element of the *bindingTemplate* XML element in the *businessEntity* structure to obtain the URL of a specific service provided by the business. Option [3] invokes the business' web site **160** with the cached URL to access the desired web pages. Since the accessed web pages from the web site **160** can contain a large quantity of information, it can be optionally cached in the cache **144** of the knowledge engine server **140** of **Figure 1**. Typically, if *bindingTemplet* is used to attempt contact services directly at the *accessPoint* and fails, the terminal may use the *bindingTemplate*'s unique ID to fetch a new *bindingTemplate* instance from the UDDI registry, assuming that the new instance contains up-to-date information on the service.

Option [4] of the Mobile Web Services menu of **Figure 1**, is:

[4] ENTER SEARCH HANDLE TO USE PROFILE FROM A PRIOR SEARCH

Option [4] enables the user to replay a prior UDDI registry search strategy. The user typically enters a previously used search handle into the wireless device and selects the replay Option [4]. The wireless device then accesses the UDDI registry search strategy from the user's stored profile corresponding to the search handle and loads the business name query, the selected *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data as each respective operand that would have been otherwise entered by the user. If the data in the

UDDI registry has been updated since the user's last query, the bindingTemplate data may contain additional or modified *accessPoint URLs*, of the selected service on the web site of the selected business. **Figure 1I** and **1J** illustrate the user's wireless device in a progression of stages as it carries out the method of replaying a UDDI registry search strategy as a shortcut for online or offline queries to the UDDI registry. **Figure 2B** discloses a flow diagram of the sequence of operational steps for the wireless device's program to replay the UDDI registry search strategy. **Figure 3B** is a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out the program to replay the UDDI registry search strategy.

The sequence of operational steps for the wireless device's UDDI registry browsing method are shown in **Figure 2A**. The Mobile Web Services menu of **Figure 1A** begins by entering a search handle that will be associated with the user's search strategy. The query terms are subsequently entered by the user, which may be a business name, part of a business name, a service description, or other characterization. If the characterization is part of a business name, the wireless device then sends a *find_business* XML inquiry using Simple Object Access Protocol (SOAP) to the UDDI registry. The wireless device then receives back from the UDDI registry, a *businessList* message shown in the Mobile Web Services menu of **Figure 1B**, that contains a list of business names satisfying the *find_business* query. The user can then select an item from the returned *businessList* message and drill-down in the selected business' entity data.

The Mobile Web Services menu of **Figure 1C** shows the wireless device then sending a *find_service* XML inquiry using SOAP to the UDDI registry. The Mobile Web Services menu of **Figure 1D** shows the wireless device then receiving back from the UDDI registry, a *serviceList* message that contains a list of names of services offered by the selected business. The user can

then select an item from the returned *serviceList* message and drill-down in the selected service data, as shown by the Mobile Web Services menu of **Figure 1E**.

The wireless device then sends a *_get_serviceDetail_XML* inquiry using SOAP to the UDDI registry, as shown by the Mobile Web Services menu of **Figure 1E**. The Mobile Web Services menu of **Figure 1F** then shows the wireless device receiving back from the UDDI registry a *serviceDetail* message that includes *bindingTemplate* data that contains the details of the selected service. Included in the *bindingTemplate* data is the *accessPoint URL*, which is the URL of the selected service on the web site of the selected business.

The Mobile Web Services menu of **Figure 1G** shows the wireless device displaying the *accessPoint URL* to the user. The Mobile Web Services menu of **Figure 1H** shows that the *serviceDetail* message can contain one or a plurality of URLs, depending on the number of matches made against the user's query in the search by the UDDI registry. The wireless device also stores the search handle in user profile with the selected *accessPoint URL*, to enable the user to access web pages from the web site of the selected business. This provides the user with a shortcut for accessing pages from web sites, in response to the user's entry of abbreviated search handle to the wireless device.

The wireless device also stores the search handle in user profile with the UDDI registry search strategy. The stored search strategy includes the business name query , the selected *businessEntity* data, the selected *businessService* data, the selected *bindingTemplate* data, and the selected *accessPoint URL*. This provides the user with a shortcut for online or offline queries to the UDDI registry, in response to the user's entry of abbreviated search handle to the wireless device.

To replay a UDDI registry search strategy, the user enters a search handle into the wireless device and selects the replay option, as shown in the Mobile Web Services menu of **Figure 1I**. The wireless device then accesses the UDDI registry search strategy from user profile corresponding to the search handle and loads the business name query, the selected
5 *businessEntity* data, the selected *businessService* data, and the selected *bindingTemplate* data as each respective operand that would have been otherwise entered by the user. If the data in the UDDI registry has been updated since the user's last query, the *bindingTemplate* data may contain additional or modified *accessPoint URLs*, of the selected service on the web site of the selected business, as shown by the Mobile Web Services menu of **Figure 1J**.

Figure 2 is a functional block diagram of the wireless device 100, and illustrates its various components and programs. **Figure 2** discloses the memory 202 connected by means of the bus 204 to the radio 206, the keypad 104, the central processor 210 and the display 212. The memory 202 contains program modules each consisting of a sequence of programmable instructions. The wireless devices UDDI registry browsing program 240 (see **Figure 2A**) is
15 stored in the memory 202. The wireless devices replay UDDI registry search strategy program 270 (see **Figure 2B**) is also stored in the memory 202. The indirect session thru server program 220 is also stored in the memory 202.

User data 222 is stored in the memory 202, which includes the user ID 230 and the user profile 232. The user profile 232 includes the user's name and email address, the user's search
20 handles, the UDDI search strategies, the sorting and filtering specifications, selected URLs, selected document titles and binding templates which contain URLs. Also contained in the memory 202 of **Figure 2** is the cache 224 wherein documents and lists returned from a website, can be stored. In addition, the WAP client program 108 is stored in the memory 202.

Figure 2a is a flow diagram disclosing the sequence of operational steps for the wireless device's UDDI registry browsing program **240**. The steps depicted in **Figure 2A** are as follows:

5 **Step 250:** WIRELESS DEVICE BROWSING UDDI REGISTRY ENTER SEARCH
HANDLE "_STOCKS_"

Step 252: ENTER QUERY TERMS "_WALL STREET_" + "_FINANC*" IS THIS A
BUSINESS NAME? Y/N "_Y_" IS THIS A SERVICE NAME? Y/N "_N_"

10 **Step 254:** SEND "_find_business_" XML INQUIRY WITH SOAP PROTOCOL TO
UDDI REGISTRY

15 **Step 256:** SELECT ITEM FROM RETURNED businessList MESSAGE DRILL-
DOWN BUSINESS ENTITY DATA "_WALL STREET JOURNAL
ONLINE_"

Step 258: SEND "_find_service_" XML INQUIRY WITH SOAP PROTOCOL TO
UDDI REGISTRY

20 **Step 260:** SELECT ITEM FROM RETURNED serviceList MESSAGE DRILL-DOWN
BUSINESS SERVICE DATA "_TECH STOCK NEWS_"

Step 262: SEND "_get_serviceDetail_" XML INQUIRY WITH SOAP PROTOCOL TO
UDDI REGISTRY

25 **Step 264:** SELECT ITEM OF RETURNED serviceDetail MESSAGE DISPLAY
accessPoint URL FROM bindingTemplate DATA "COMPUTERS" IN
RETURNED serviceDetail MESSAGE URL =
"_www.wsj.com/news/computers_"

30 **Step 266:** STORE SEARCH HANDLE "_STOCKS_" IN USER PROFILE WITH
SELECTED URL = "_www.wsj.com/news/computers_"

35 **Step 268:** STORE SEARCH HANDLE "_STOCKS_" IN USER PROFILE WITH UDDI
REGISTRY SEARCH STRATEGY: BUSINESS NAME QUERY "_WALL
STREET_" + "_FINANC*" businessEntity DATA SELECTED "_WALL
STREET JOURNAL ONLINE_" businessService DATA SELECTED
"_TECH STOCK NEWS_" bindingTemplate DATA SELECTED
"_COMPUTERS_" accessPoint URL SELECTED
40 "_www.wsj.com/news/computers_"

Figure 2B shows a flow diagram of the sequence of operational steps for the wireless device's program to replay the UDDI registry search strategy. **Figure 2B** includes the following steps:

Step 271: REPLAY UDDI REGISTRY SEARCH STRATEGY ENTER SEARCH HANDLE “_STOCKS_”.

Step 272: ACCESS UDDI REGISTRY SEARCH STRATEGY IN USER PROFILE FOR SEARCH HANDLE “_STOCKS_”, BUSINESS NAME QUERY “_WALL STREET_” + “_FINANC_”, businessEntity DATA SELECTED “_WALL STREET JOURNAL ONLINE_”, businessService DATA SELECTED “_TECH STOCK NEWS_”, bindingTemplate DATA SELECTED “_COMPUTERS_”, accessPoint URL SELECTED “_www.wsj.com/news/computers_”.

Step 274: SEND *_find_business_* SML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY.

Step 276: SELECT ITEM FROM RETURNED *businessList* MESSAGE DRILL-DOWN BUSINESS ENTITY DATA “_WALL STREET JOURNAL ONLINE_”.

Step 278: SEND *_find_service_* XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY.

Step 280: SELECT ITEM FROM RETURNED *serviceList* MESSAGE DRILL-DOWN BUSINESS SERVICE DATA “_TECH STOCK NEWS_”.

Step 282: SEND *_get_serviceDetail_* XML INQUIRY WITH SOAP PROTOCOL TO UDDI REGISTRY.

Step 284: SELECT ITEM OF RETURNED *serviceDetail* MESSAGE DISPLAY accesPoint URLs FROM bindingTemplate DATA “COMPUTERS” IN RETURNED *serviceDetail* MESSAGE, URL = “_www.wsj.com/news/computers_”, URL = “_www.wsj.com/quotes/computers_”, URL = “_www.wsj.com/graphs/computers_”.

Figure 3A discloses a network process flow diagram showing the interaction of the wireless device and the UDDI registry when carrying out the UDDI registry browsing program of **Figure 2A**. The network process flow diagram in **Figure 3A** consists of three columns labeled respectively, wireless device **100**, server **140** and UDDI registry **170**. The process flow diagram **Figure 3A** illustrates the interaction between steps carried in the wireless device **100** and steps carried out in the UDDI registry **170**. The diagram of **Figure 3A** begins with the wireless device **100** step **250** where a search handle is entered. Then, at step **252**, query terms are entered. At step **254**, the *_find_business_* query is sent to the UDDI registry **170**. In the UDDI registry column **170** in **Figure 3A**, the UDDI registry conducts searches based on the *_find_business_* query and returns the *businessList* in step **255**. The flow then returns to the wireless device **100** and passes to step **256** wherein an item of the *businessList* is selected which typically is a *businessEntity*. The *businessEntity* is then drilled down. The flow then passes to step **258** in which the *_find_service_* query is sent to the UDDI registry. The flow then passes to the UDDI registry **170** where the *_find_service_* query is searched and the service list is returned in step **259**. Then the flow passes to the wireless device **100** where an item of the *serviceList* is selected which is a *businessService*. The *businessService* is subsequently drilled down in step **260**. The flow proceeds to step **262** in which the *_get_serviceDetail_* query is sent to the UDDI registry. Then the flow passes to the UDDI registry **170** where the *_get_serviceDetail_* query is responded to and the binding template is returned in step **263**. Then the flow passes back to the wireless device **100** where in step **264**, the *serviceDetail* is selected from the *bindingTemplate* and the accessPoint URL is stored in step **264**.

Figure 3B illustrates a network process flow diagram of the interaction of the wireless device and the UDDI registry when carrying out a program to replay the UDDI registry search

strategy. **Figure 3B** is divided into three columns, the wireless device column **100**, the server **140** column, and the UDDI registry **170** column. **Figure 3B** starts with the wireless device entering the replay UDDI registry search strategy wherein the search handle is entered in step **271**. Then the process flows to step **272** where the search strategy is accessed in the user profile which corresponds to the search handle which was input in step **271**. **Figure 3B** then proceeds through the remainder of the processes in a similar manner as that disclosed in **Figure 3A**.

Figure 3B discloses how the user is enabled to replay a prior UDDI registry search strategy. The user merely enters a previously used search handle into the wireless device and selects the replay option. The wireless device then accesses the UDDI registry search strategy from the user's stored profile corresponding to that search handle. This may be performed either at the wireless device **100** or, in the alternate embodiment in the server **140**. The search strategy includes information such as the *businessEntity* data and *businessService* data and *bindingTemplate* data that was found during the course of an earlier search of the UDDI registry **170**. This data contained in the UDDI registry search strategy (accessed from the stored profile) is then loaded as the data for each respective operand that would have been otherwise entered by the user. In this way, the flow diagram of **Figure 3B** enables the user to automatically invoke a search strategy of the UDDI registry **170**, that was used in the past.

Figure 4 is a functional block diagram of the knowledge engine server, showing the memory storing the application services software programs needed to perform the operations of an embodiment of the invention. **Figure 4** discloses the functional components of an exemplary knowledge engine server **140** arranged as an object model. The object model groups the object oriented software programs into components that perform the major functions and applications in knowledge engine server **140**. The object model for memory **402** of knowledge engine server

140 employs a three-tier architecture that includes presentation tier 415, infrastructure objects partition 422, and business logic tier 414. The object model further divides business logic tier 414 into two partitions, application objects partition 422 and data objects partition 426.

Presentation tier 415 retains the programs that manage the device interfaces to knowledge engine server 140. In Figure 4, presentation tier 415 includes network interface 420. A suitable implementation of presentation tier 415 may use Java servlets to interact with WAP protocol gateway 120 via the hypertext transfer protocol ("HTTP"). The Java servlets run within a request/response server that manages the exchange of messages between WAP protocol gateway 120 and knowledge engine server 140. A Java servlet is a Java program that runs within a Web server environment. A Java servlet takes a request as input, parses the data, performs logic operations, and issues a response back to WAP protocol gateway 120. The Java runtime platform pools the Java servlets to simultaneously service many requests. Network interface 420 accepts request messages from WAP protocol gateway 120 and passes the information in the request to visit object 428 for further processing. Visit object 428 passes the result of that processing to network interface 420 for transmission back to the WAP protocol gateway 120. Network interface 420 may also use network adapter 406 to exchange data with another user device.

Infrastructure objects partition 422 retains the programs that perform administrative and system functions on behalf of business logic tier 414. Infrastructure objects partition 422 includes operating system 425, and an object oriented software program component for database server interface 430, and system administrator interface 432.

Business logic tier 414 in Figure 4 includes multiple instances of visit object 428, 428', 428". A separate instance of visit object 428 exists for each network interface 420 session. Each

visit object 428 is a stateful session bean that includes a persistent storage area from initiation through termination of the session, not just during a single interaction or method call. The persistent storage area retains information associated with the session.

When WAP protocol gateway 120 sends a query message to knowledge engine server 140, the message is sent to network interface 420 to invoke a method that creates visit object 428 and stores connection information as a state in visit object 428. Visit object 428 may, in turn, invoke a method in UDDI registry browsing application 440 to browse the UDDI registry 170. Application 440 sends queries to the UDDI registry, as discussed above.

When WAP protocol gateway 120 sends a search handle message to knowledge engine server 140, a message is sent to network interface 420 to invoke a method that creates visit object 428 and stores connection information as a state in visit object 428. Visit object 428 may, in turn, invoke a method in replay UDDI registry search strategy application 442 to replay a prior search strategy. The application 442 performs the replay method discussed above. Similar operations occur for applications 444, 446 and 448 in Figure 4.

Figure 4A is a more detailed functional block diagram of the server, showing the knowledge engine 142. The knowledge engine 142 includes a program which is shown in Figure 4A as a sequence of steps 1 thru 13.

KNOWLEDGE ENGINE 142

- [1] RECEIVE USER'S QUERY OR SEARCH HANDLE
- [2] IF SEARCH HANDLE RECEIVED, THEN REPLAY UDDI SEARCH PROFILE & GOTO 6
- [3] ACCESS UDDI REGISTRY WITH USER'S QUERY TO IDENTIFY WEB SITES
- [4] RETURN LIST OF WEB SITES TO USER AND STORE BINDING TEMPLATES IN PROFILE

- [5] RECEIVE USER'S SELECTION OF WEB SITES TO SEARCH & UPDATE USER PROFILE
[6] SEARCH THE IDENTIFIED WEB SITES USING URLS FROM BINDING TEMPLATES
5 [7] RETRIEVE DOCUMENTS RESULTING FROM SEARCH
[8] SORT LIST OF DOCUMENTS IN ACCORDANCE WITH USER'S PROFILE
[9] STORE DOCUMENTS AND LIST IN CACHE
[10] RETURN LIST OF DOCUMENTS TO USER WHENEVER USER IS LOGGED ON
10 [11] RECEIVE USER'S SELECTIONS FROM LIST AND UPDATE USER'S PROFILE
[12] RETURN SELECTED DOCUMENTS TO USER
[13] ASSOCIATE SEARCH HANDLE WITH USER'S SELECTIONS IN USER'S PROFILE

15 Also provided in server 140 is the user data 146 which includes the user ID profile 230
which is discussed above. Since a plurality of users may make use of the server 140, there are a
plurality of user profiles shown in **Figure 4A**, one for the user ID 230' having user profile 232'
and another for the user ID 230" having user profile 232". The server 140 of **Figure 4A** also has
a cache 144 which stores documents and lists which are obtained from the various websites 160
that have been interrogated by the user with the aid of the server 140.
20

Figure 4B is a flow diagram of the sequence of operational steps for the server 140 UDDI
registry browsing program 170. **Figure 4B** has three columns, the first column labeled user's
wireless device 100, the second column labeled server 140, and a third column labeled UDDI
registry 170 and web sites 160. **Figure 4B** illustrates the interaction of the wireless device 100,
25 the server 140, the UDDI registry 170 and the web sites 160, in accordance with an embodiment
of the invention. Starting with the user's wireless device 100, **Figure 4B** shows sending a query
to the server, in step 302. At the server 140, the query is received from the user in step 304, and
the process flows to step 306 where web sites are identified from the UDDI registry and the
user's profile is updated. The process in step 306 for identification of the web sites from the

UDDI registry is the process which has been discussed above in connection with **Figures 2A and 3A**. The process then flows to step **330** in the UDDI registry **170**, wherein the UDDI registry accesses the requested information in response to the queries sent from the server **140** to identify web sites. Step **330** then transfers the results of that search from the UDDI registry **170** back to the server **140**. At the server **140**, the process flows to step **314** wherein the server has taken the information identifying the web sites received from the UDDI registry **170**, and formulates a request to retrieve documents which is sent to the web sites **160**. The process then flows to step **332** where the web sites **160** receiving the request from the server **140**, access their respective servers for the requested documents and then return the documents to the server **140**. The server **140** then sorts the documents into a list in accord with the user's profile, sorting the list into the order requested by the user, and filtering out any documents which the user is not interested, in accordance with the user's profile. The process then flows to step **316** in which the documents are stored in the cache at the server, cache **144**, and the list which has been sorted by the server **140**, is returned to the user. The process then flows to step **320** at the user's wireless device **100** where the sorted list received from the server **140** is presented to the user and the user can select from that list those documents desired to be reviewed. In step **320**, the user's request for documents is then sent back to the server **140**. The process then flows to step **326** where the server **140** accesses its cache **144** to retrieve those documents selected by the user in step **320**, and then the server **140** returns the selected documents to the user's wireless device **100**. Step **326** then compiles the user's preferences in the user profile **230**. The server **140** can also update the user's preferences in the user's profile in step **328**. The process flows from step **326** to step **322** at the user's wireless device **100**, where the user receives the selected documents.

Although illustrative embodiments have been described herein in detail, it should be noted and understood that the descriptions and drawings have been provided for purposes of illustration only and that other variations both in form and detail can be made thereupon without departing from the spirit and scope of the invention. The terms and expressions have been used
5 as terms of description and not terms of limitation. There is no limitation to use the terms or expressions to exclude any equivalents of features shown and described or portions thereof.

10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2